AF/SPW

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 200309108-1

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Carlos A. BONILLA	**Confirmation No.:** 7122
Application No.: 10/685,990	**Examiner:** Lau, T. S.
Filing Date: 10/14/03	**Group Art Unit:** 2863

Title: AUTOMATIC SOFTWARE TESTING

Mail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on 08/03/06.

The fee for filing this Appeal Brief is (37 CFR 1.17(c)) $500.00.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

☐ (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d)) for the total number of months checked below:

☐ 1st Month $120	☐ 2nd Month $450	☐ 3rd Month $1020	☐ 4th Month $1590

☐ The extension fee has already been filed in this application.

☒ (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account 08-2025 the sum of $ 500. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.

☒ I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450

Date of Deposit: 10/03/06

OR

☐ I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number (571)273-8300.

Date of facsimile:

Typed Name: Desiree Reardon

Signature:

Respectfully submitted,

Carlos A. BONILLA

By

John P. Wagner, Jr.

Attorney/Agent for Applicant(s)

Reg No. : 35,398

Date : 10/03/06

Telephone : (408) 938-9060

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Appellant:	Bonilla	Patent Application	
Serial No.:	10/685,990	Group Art Unit:	2863
Filed:	October 14, 2003	Examiner:	Lau, Tung S.

For: AUTOMATIC SOFTWARE TESTING

Appeal Brief

Table of Contents

Real Party in Interest

The assignee of the present invention is Hewlett-Packard Company.

Related Appeals and Interferences

There are no related appeals or interferences known to the Appellant.

Status of Claims

Claims 1-20 have been rejected. This appeal involves Claims 1-20.

Status of Amendments

All proposed amendments have been entered. An amendment subsequent to the Final Action has not been filed.

Summary of Claimed Subject Matter

Independent Claims 1, 8 and 14 of the present application pertain to various embodiments for automatic software testing. For example, Claim 14 recites,

> "A computer system for automatic software testing comprising:
> a test master computer system;
> a plurality of test computer systems communicatively coupled to said test master computer system;
> a common information point communicatively coupled to said test master computer system and to said plurality of test computer systems;
> said test master computer system for installing a test driver on each of said plurality of test computer systems;
> at least one of said plurality of test computer systems for:
> storing status information of a software test running on said at least one of said plurality of test computer systems to said common information point;
> automatically reinstalling an operating system on said at least one of said plurality of test computer systems;
> querying said common information point to determine said status information; and
> resuming said software test."

The following description of Claim 14 shall refer to page 8 line 3 to page 13 line 10 and refer to Figure 1 described on page 13 line 12 to page 15 line 5 of the instant application serial no. 10/685,990. Referring to page 13 lines 14-15, according to one embodiment, a test environment 100 includes a test management computer system 110 and a plurality of test computer systems 120, 130, and 140. Referring to page 13 lines 16-18, the test computer systems 120-140 are communicatively coupled to the test management computer system 110.

Referring to page 13 line 20, the test environment 100 further comprises a common information point 115. Referring to page 13 lines 24-25, the common information point 115 is communicatively coupled to the test management

computer system 110 and the test systems 120-140. Refer to page 10 line 13 to page 11 line 25 for more information on the common information point.

Referring to page 14 lines 6-7, the test computer system 110 can install (111) a test driver on test system 120. Page 9 lines 12-13 state, "The test master process installs a test driver on the test system." Referring to page 14 lines 13-14, communication 125 indicates test system 120 stores information of its test status to common information point 115.

Referring to page 8 lines 22-24, the instant application states that at times, "it may be desirable to reinstall an operating system to establish a known testing environment prior to continuing testing." Page 9 lines 1-2 of the instant application also state, "...it may be desirable to change operating systems in a controlled and automated manner during testing..."

Referring to page 12 lines 1-2 of the instant application state, "A test driver queries the common information point to determine what, if any, tests to run." Referring to page 12 lines 1-2 the instant application states, "A test driver queries the common information point to determine what, if any, tests to run."

Independent Claim 1 recites,

"A computer implemented method of automatic software testing comprising:
storing status information of a software test running on a test system to a common information point;
automatically reinstalling an operating system on said test system;

querying said common information point to determine said status information; and
resuming said software test."

Referring to Figure 3 and page 16 line 23 to page 17 line 21, the instant application describes storing status information for block 310, automatically reinitializing an operating system for block 320, querying said common information point at block 330 and resuming the software test at block 340.

Claim 8 recites,

"A computer implemented method of automatic software testing comprising:
installing test driver software on a plurality of test systems;
providing a mapping of a plurality of virtual test system names to real test system names to said test driver software; and
gathering test results from said plurality of test systems."

Referring to Figure 4 and page 18 line 20 to page 19 line 17, the instant application describes installing test driver software at block 410, providing a mapping of a plurality of virtual test system names to real test system names at block 420, and gathering test results at block 430.

Claims 2-7 depend on Claim 1. Claims 9-13 depend on Claim 8. Claims 15-20 depend on Claim 14. These dependent claims include all of the limitations of their respective independent claims and include additional limitations. Therefore these dependent claims should be patentable for at least the reasons that their respective independent claims should be patentable.

Grounds of Rejection to be Reviewed on Appeal

1. In paragraph 1a on page 2 of the Final Office Action dated May 30, 2006, Claims 1-7 and 14-20 are rejected under 35 U.S.C. §102(b) as being anticipated by U.S. patent no. 6,317,845 by Meyer et al. (referred to hereinafter as "Meyer").

2. In paragraph 1b on page 6 of the Final Office Action dated May 30, 2006, Claims 8-13 are rejected under 35 U.S.C. §102(a) as being unpatentable over U.S. patent publication no. 2003/0051186 by Boudnik et al. (referred to hereinafter as "Boudnik").

Arguments

1. Whether Claims 1-7 and 14-20 are anticipated by Meyer (6,317,845) under 35 U.S.C. 102(b).

A. Scope and Content of the Cited Prior Art Reference (Meyer)

At Col. 2 lines 25-28 of the background section, Meyer states that the problem with prior art disaster recovery is that the bootable floppy disks used for disaster recovery have relatively low capacity. Meyer also states at Col. 2 lines 12-16 that another problem with conventional disaster recovery is that users are required to remember additional commands and that there is limited documentation and on-line help.

Meyer solves the problem of a bootable floppy disk having limited capacity by teaching a way for a user to restart a computer by manually inserting a removable high capacity disk into the computer. For example, Meyer states in the abstract, "In the event that a user encounters an abnormal operating condition, the user inserts the removable high capacity disk into the computer and restarts the computer." Some other places where Meyer teaches a user manually restarting a computer by inserting a removable high capacity disk are Figure 2 step 104, Col. 3 lines 23-24, Col. 10 lines 60-61, and Col. 12 lines 22-32.

Meyer solves the problem of requiring users to remember additional commands and limited documentation and on-line help by incorporating a general user interface on his removable high capacity disk that guides the user

through the recovery process. For example, at Col. 13 lines 8-11, Meyer states, "Referring back to FIG. 2, once the GUI is launched the user is prompted to proceed with the recovery process at step 108. ... The user is prompted to run simple, easy to-use restore and rescue applications within the operating environment of the graphical user interface." Some other portions where Meyer teaches the GUI are Figure 2 steps 108 and 112, Col. 12 lines 57-58, and Col. 13 lines 37-39.

B. Differences Between the Cited Prior Art References and the Claimed Invention.

Meyer does not teach or suggest, "storing status information of a software test running on a test system to a common information point," as recited by Claim 1. Meyer teaches "recovery" not "testing." Meyer does not teach anything about "status information of a software test running on a test system" let alone using a "common information point." Since Meyer teaches recovery instead of "testing," Meyer would have no need for teaching a "common information point."

The Office Action asserts that Meyer teaches "storing status information of a software test running on a test system to a common information point" in the abstract. Appellant is uncertain what in Meyer's abstract the Office Action is referring to. For the sake of argument, Appellant shall assume that the Office Action is referring to the abstract's statement, "The removable high capacity disk also includes a suite of software recovery software which attempt to ascertain and correct the cause of the abnormal operating condition to return the computer system to a normal operating condition." However, this portion of

the abstract does not teach "storing," "a software test," or "a common information point," let alone teach or suggest "storing status information of a software test running on a test system to a common information point."

The Office Action also asserted that Meyer teaches "storing status information of a software test running on a test system to a common information point" at unit 108 of FIG. 2. Meyer discusses unit 108 in the second paragraph of Col. 13. However, the second paragraph of Col. 13 in Meyer discusses recovery applications but fails to teach or suggest "software test," "common information point," and "storing" let alone teach or suggest "storing status information of a software test running on a test system to a common information point."

Meyer does not teach or suggest, "automatically reinstalling an operating system on said test system," as recited by Claim 1. In fact, Meyer teaches away from "automatically reinstalling an operating system" (emphasis added) since Meyer requires user involvement in order to "restore" an operating system. As already stated Meyer requires the use of a GUI so that users are not required to remember additional commands and to address the problem of limited documentation and on-line help.

The Office Action asserts that Meyer teaches "automatically reinstalling an operating system on said test system," in the abstract. However, Meyer makes no mention of a test system in the abstract. Further, Meyer requires that a user manually insert a removable high capacity disk into the computer and manually restart the computer. The Office Action also asserts that Meyer teaches

"automatically reinstalling an operating system on said test system," with unit 106 depicted in FIG. 2. Meyer discusses unit 106 at Col. 12 lines 20-47. In Col. 12 lines 20-47 Meyer states several times that the user inserts the high capacity disk and the user restarts the computer. Therefore, Meyer does not teach or suggest "automatically reinstalling an operating system on said test system."

Meyer does not teach or suggest, "querying said common information point to determine said status information," as recited by Claim 1. Since Meyer does not teach a "common information point" or "status information" then Meyer cannot teach or suggest "querying said common information point to determine said status information."

The Office Action asserts that Meyer teaches "querying said common information point to determine said status information" in the abstract and with unit 110 of FIG. 2. However, as already explained herein Meyer fails to teach "a common information point" in the abstract. Meyer discusses step 110 at Col. 13 lines 37-40 which states, "If at step 110 it is determined that the error has been corrected, then at step 112, the user is prompted to restart the computer in the normal manner (i.e., boot from the hard drive)." As can be seen, Col. 13 lines 37-40 say nothing about "querying said common information point to determine said status information." Further, since Meyer fails to teach "storing status information of a software test running on a test system to a common information point" Meyer cannot teach or suggest "querying said common information point to determine said status information."

Further, Meyer does not teach or suggest, "resuming said software test," as recited by Claim 1. Since Meyer does not teach "software test," Meyer cannot teach "resuming said software test." The Office Action asserts that Meyer teaches "resuming said software test," in the abstract and at unit 116 of FIG. 2. However, Meyer does not teach "software test" anywhere let alone teach "resuming said software test."

In the response to arguments section, the Office Action asserts that Meyer discloses "storing...," as recited by Claim 1 in the abstract, Fig. 2, unit 108, and unit 110. The difference between "storing...," as recited by Claim 1 and Meyer's abstract and Meyer's unit 108 have already been discussed herein. The difference between "storing ...," as recited by Claim 1 and Meyer's unit 110 is that Meyer's unit 110 is a decision box for determining whether an error has been corrected rather than "storing ...," as recited by Claim 1.

In the response to arguments section, the Office Action also asserted that Meyers teaches "automatically installing..." at Col. 12 lines 48-62 and Col. 13 lines 8-11. Col. 12 lines 48-62 clearly indicates that a graphical user interface is launched and therefore does not teach "automatically reinstalling an operating system..." (emphasis added) as recited by Claim 1. Col. 13 lines 8-11 state, "Referring back to FIG. 2, once the GUI is launched the user is prompted to proceed with the recovery process at step 108. Alternatively, the recovery process may proceed automatically after the computer is booted." However, Col. 13 lines 8-11 does not teach "automatically reinstalling an operating system..." (emphasis added) as recited by Claim 1.

In the response to arguments section, the Office Action also asserted that Meyers teaches "querying...," as recited by Claim 1 in the abstract and at unit 110 of Fig. 2. The difference between "querying...," as recited by Claim 1 and Meyer's abstract and Meyer's unit 110 of Fig. 2 have already been discussed herein.

For the foregoing reasons, independent Claim 1 should be patentable over Meyer in that Meyer is missing essential elements, "storing status information of a software test running on a test system to a common information point; automatically reinstalling an operating system on said test system; querying said common information point to determine said status information; and resuming said software test," and therefore the anticipation rejection of Claim 1 under §102(b) is improper and should be reversed.

Independent Claim 14 recites limitations that are similar to "storing..., automatically reinstalling... querying... resuming...," which are recited by Claim 1. Therefore, independent Claim 14 should be patentable over Meyer for similar reasons that Claim 1 should be patentable over Meyer.

Claims 2-7 depend on Claim 1. Claims 15-20 depend on Claim 14. These dependent claims include all of the limitations of their respective independent claims and include additional limitations. Therefore these dependent claims should be patentable for at least the reasons that their respective independent claims should be patentable.

2. Whether Claims 8-13 are anticipated by Boudnik (2003/0051186) under 35 U.S.C. 102(a).

A. Scope and Content of the Cited Prior Art References (Boudnik)

Referring to the middle of paragraph 0029, among other places, Boudnik teaches the use of a Java virtual machine. Referring to the last half of paragraph 0054, Boudnik only teaches the use of one virtual machine, a Java virtual machine. Further, in paragraph 0054 Boudnik teaches locating "an available test system...to execute each of the test execution requests 116a-116c."

B. Differences Between the Cited Prior Art References and the Claimed Invention.

Boudnik does not teach or suggest, "installing test driver software on a plurality of test systems," as recited by Claim 8. For example, referring to paragraph 0054 Boudnik teaches locating "an available test system ... to execute each of the test execution requests 116a-116c" and therefore teaches away from "installing test driver software on a plurality of test systems."

Although Boudnik teaches the use of a Java virtual machine, Boudnik does not teach "providing a mapping of a plurality of virtual test system names to real test system names to said test driver software," as recited by Claim 8. Boudnik only teaches the use of one virtual machine, a Java virtual machine. Further, since Boudnik teaches locating "an available test system" rather than "installing test driver software...," Boudnik would have no motivation to teach a "mapping...," as recited by Claim 8. Therefore Boudnik does not teach

"providing a mapping of a plurality of virtual test system names to real test system names to said test driver software," as recited by Claim 8.

The Office Action asserts that Boudnik teaches "providing a mapping of a plurality of virtual test system names to real test system names to said test driver software" at FIG. 6, Fig. 5, unit 500, FIG. 8, unit 812. However, FIG. 6 depicts a post mortem object which includes test suite names 600, work directory name 602, result directory name 604, point of execution 606, and system name 608. Among other things, there is nothing in FIG. 6 about virtual test system names. Therefore, FIG. 6 cannot teach or suggest "providing a mapping of a plurality of virtual test system names to real test system names to said test driver software."

Boudnik's FIG. 5 depicts a test system 114 that includes an agent process and a test harness 502. The agent process 120 communicates with a post mortem object 508 and a system controller 108. However, FIG. 5 depicts nothing about a mapping let alone anything that teaches or suggests "providing a mapping of a plurality of virtual test system names to real test system names to said test driver software."

Unit 812 of FIG. 8 is discussed in paragraph 0084 of Boudnik. However, there is nothing in paragraph 0084, which discusses operation 812, about "providing a mapping of a plurality of virtual test system names to real test system names to said test driver software."

In the response to arguments section, the Office Action states, "Boudnik clearly discloses 'providing a mapping of a plurality of virtual system names to

real test system names to test driver software' in fig. 4, 5, 6, 8, 9, where Boudnik uses Java (virtual software) to identify (mapping) each objects (fig. 4, unit 906) and each object is link to hardness (real system information) information list on fig. Fig. 5 and 4." As already stated, FIGS. 5, 6, and 8 do not teach or disclose "mapping of a plurality of virtual test system names to real test system names." Paragraph 0065 states concerning FIG. 4 "The test configuration 400 includes a test suite comprising a test list 402 having a plurality of individual tests 404." Therefore FIG. 4 does not teach or suggest "mapping of a plurality of virtual test system names to real test system names." In regards to operation 902, paragraph 0090 of Boudnik states, "In operation 902, the agent process refers to the JavaSpace of the system.....JavaSpaces technology provides developers with the ability to create and store objects with persistence, which allows for process integrity." Therefore, paragraph 0090 does not teach or suggest "mapping of a plurality of virtual test system names to real test system names" either.

For the foregoing reasons independent Claim 8 should be patentable over Boudnik in that Boudnik is missing essential elements, "installing test driver software on a plurality of test systems; providing a mapping of a plurality of virtual test system names to real test system names to said test driver software," and therefore the anticipation rejection of independent Claim 8 under §102(a) is improper and should be reversed.

Claims 9-13 depend on Claim 8. These dependent claims include all of the limitations of their respective independent claims and include additional

limitations. Therefore these dependent claims should be patentable for at least the reasons that their respective independent claims should be patentable.

In summary, the Appellant respectfully requests that the Board reverse the Examiner's rejections of Claims 1-20.

The Appellant wishes to encourage the Examiner or a member of the Board of Patent Appeals to telephone the Appellant's undersigned representative if it is felt that a telephone conference could expedite prosecution.

Respectfully submitted,

WAGNER, MURABITO & HAO LLP

Date: 10/2/06

John P. Wagner

Registration Number: 35,398

WAGNER, MURABITO & HAO LLP
Westridge Business Park
123 Westridge Drive
Watsonville, CA 95076
408-938-9060

Claims Appendix

1. (Previously Presented) A computer implemented method of automatic software testing comprising:

storing status information of a software test running on a test system to a common information point;

automatically reinstalling an operating system on said test system;

querying said common information point to determine said status information; and

resuming said software test.

2. (Original) The method of Claim 1 wherein said common information point is on a computer system independent from a computer system running said software test.

3. (Previously Presented) The method of Claim 1 wherein said reinstalling is performed under software control.

4. (Original) The method of Claim 1 wherein said querying is started by a start-up process, said start up process automatically initiated by said operating system.

5. (Original) The method of Claim 1 wherein said status information comprises an identification test portions completed.

6. (Original) The method of Claim 5 wherein said resuming restarts said software test at point subsequent to a last test portion completed.

7. (Previously Presented) The method of Claim 1 wherein said test system is running a different operating system subsequent to said reinstalling than said test system was running prior to said reinstalling.

8. (Original) A computer implemented method of automatic software testing comprising:

installing test driver software on a plurality of test systems;

providing a mapping of a plurality of virtual test system names to real test system names to said test driver software; and

gathering test results from said plurality of test systems.

9. (Original) The method of Claim 8 further comprising identifying a common point of information to said plurality of test systems.

10. (Original) The method of Claim 9 wherein said test results are gathered from said common point of information.

11. (Original) The method of Claim 9 wherein said mapping resides on said common point of information.

12. (Original) The method of Claim 9 wherein said common point of information is a network file system mount point common to all test systems.

13. (Original) The method of Claim 9 wherein said installing is responsive to a start up process, said start up process automatically initiated by said operating system.

14. (Previously Presented) A computer system for automatic software testing comprising:

a test master computer system;

a plurality of test computer systems communicatively coupled to said test master computer system;

a common information point communicatively coupled to said test master computer system and to said plurality of test computer systems;

said test master computer system for installing a test driver on each of said plurality of test computer systems;

at least one of said plurality of test computer systems for:

storing status information of a software test running on said at least one of said plurality of test computer systems to said common information point;

automatically reinstalling an operating system on said at least one of said plurality of test computer systems;

querying said common information point to determine said status information; and

resuming said software test.

15. (Original) The computer system of Claim 14 wherein said test master computer system is distinct from said plurality of test computer systems.

16. (Original) The computer system of Claim 14 wherein said reinitializing is performed under software control.

17. (Original) The computer system of Claim 14 wherein said querying is started by a start up process, said start up process automatically initiated by said operating system.

18. (Original) The computer system of Claim 14 wherein said status information comprises an identification of test portions completed.

19. (Original) The computer system of Claim 18 wherein said resuming starts said software test at point subsequent to a last test portion completed.

20. (Previously Presented) The computer system of Claim 14 wherein said at least one of said plurality of test computer systems is running a different operating system subsequent to sad reinstalling than said at least one of said plurality of test computer systems was running prior to said reinstalling.

Evidence Appendix

None

Related Proceedings Appendix

None